



Vivante ISP PQ Tuning Tools 6.2.23.5 User Guide

Document Revision 1.10
31 Oct 2023

Legal Notice

COPYRIGHT INFORMATION

This document contains proprietary information of Vivante Corporation and VeriSilicon Holdings Co., Ltd. They reserve the right to make changes to any products herein at any time without notice and do not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by Vivante and/or VeriSilicon; nor does the purchase or use of a product from Vivante or VeriSilicon convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of Vivante, VeriSilicon or third parties.

DISCLOSURE/RE-DISTRIBUTION LIMITATIONS

The information contained in this Vivante proprietary copyright document may be re-distributed by Vivante customer SoC vendors who have licensed the related core IP, with the understanding that the material is re-packaged/re-branded as a licensee (SoC vendor) written/labeled document. Those portions of the SoC vendor document adapted from Vivante material should contain a notice similar to: "This section describes the Vivante® ISP and contains copyright material disclosed with permission of VeriSilicon®." (VeriSilicon Distribution **LEVEL A: PROPRIETARY INFORMATION**)

TRADEMARK ACKNOWLEDGMENT










VeriSilicon® and the VeriSilicon logo design are the trademarks or the registered trademarks of VeriSilicon Holdings Co., Ltd. Vivante® is a registered trademark of Vivante Corporation. All other brand and product names may be trademarks of their respective companies.

For our current distributors, sales offices, design resource centers, and product information, visit our web page located at <http://www.verisilicon.com>.

Vivante and VeriSilicon Proprietary. Copyright © 2023 by Vivante Corporation and VeriSilicon Holdings Co., Ltd. All rights reserved.



Table of Contents

LEGAL NOTICE	2
TABLE OF CONTENTS.....	4
LIST OF FIGURES.....	6
1 ISP PQ TUNING TOOLS OVERVIEW	8
1.1 Component Overview	8
1.2 Operating System Compatibility	8
1.3 Terminology	8
2 REQUIREMENTS	9
2.1 Software Requirements	9
2.2 Software Release Package.....	9
3 SERVER USAGE	10
3.1 Setup Environment on FPGA.....	10
3.2 Run the Server.....	12
4 VTUNER USAGE	13
4.1 General Steps	13
4.2 Start Using VTuner IDE	13
4.2.1 Install and Uninstall VTuner IDE.....	13
4.2.2 VTuner IDE for Linux	13
4.2.3 VTuner IDE for Windows.....	14
4.2.4 Using Workspaces.....	14
4.2.5 Switch Perspective	15
4.2.6 Create ISP Project	16
4.3 Common Features	17
4.3.1  Import Tuning Data	19
4.3.2  Export Tuning Data	20
4.3.3  Capture	20
4.3.4  Raw Image Input	23
4.3.5  Live Preview	24
4.3.6  Edit Register	26
4.3.7  Server Connection.....	27
4.3.8  Preferences	28
4.3.9  Extra Component.....	29
4.3.10 Project Explorer	29
4.3.11 Features View	30
4.3.12 Detail View	30
4.3.13 Console	32
4.4 Parameter Data Types.....	32
4.4.1 Boolean Value	32
4.4.2 Integer or Float Value	32
4.4.3 Option Value	33
4.4.4 Matrix Value.....	33
4.4.5 Curve Value	34
4.4.6 Color Matrix	37
4.5 AWB/AE Tool	38

4.5.1	AWB Tool	38
4.5.2	AE Tool	40
4.6	Data Sync with Server	41
4.7	Customize Sub-module and Parameters	41
4.8	Modify Sensor Work Mode	42
4.9	Customize TCP Timeout Setting	42
DOCUMENT REVISION HISTORY		43



List of Figures

Figure 1. VLC URL	11
Figure 2. Select a Workspace for VTuner IDE	14
Figure 3. Open Perspective → Other	15
Figure 4. Open Perspective → ISP Tuner	16
Figure 5. Select ISP Project Wizard	16
Figure 6. Create an ISP Project	17
Figure 7. ISP Tuner Perspective	18
Figure 8. Select a Calibration JSON File	19
Figure 9. Popup for Waiting for Import	19
Figure 10. Export Calibration and Tuning Data	20
Figure 11. Popup for Waiting for Export	20
Figure 12. Configuration for Capturing Images	21
Figure 13. Configure the Save Path in Capture Settings	22
Figure 14. Image Input Dialog	23
Figure 15. Live Preview Dialog	24
Figure 16. Edit Register	26
Figure 17. HTTP/TCP Server Configuration Dialog	27
Figure 18. UART Server Configuration Dialog	28
Figure 19. Light Indicating Server Connection Status	28
Figure 20. Preferences Setting	28
Figure 21. JSON/XML Edit	29
Figure 22. Project Explorer	29
Figure 23. Features View for the Current Project	30
Figure 24. Detail View	30
Figure 25. Control Bar	31
Figure 26. Back to 2, Undo Button and Redo Button is Enabled	31
Figure 27. Input 5, Redo Button is Disabled	31
Figure 28. Console with Messages for Selecting a Project	32
Figure 29. Boolean Value in Detail View	32
Figure 30. Integer or Float Value	32
Figure 31. Option Value	33
Figure 32. Manual UI	33
Figure 33. Auto UI	33
Figure 34. Matrix Value	33
Figure 35. Table for Editing Number Matrix Value	33
Figure 36. Table for Editing ROI Matrix Value	34
Figure 37. Curve Value: a Curve and a Table	34
Figure 38. Configure Graph Settings: Graph	35
Figure 39. Configure Graph Settings: Axes	35
Figure 40. Configure Graph Settings: Traces	36
Figure 41. Configure Graph Settings: Annotation	36
Figure 42. Graph: Annotation	37
Figure 43. Graph: Annotation	38
Figure 44. Captured Image and 32 * 32 Blocks	38
Figure 45. Select a Block	39
Figure 46. AWB Statistics Info and Chart	39
Figure 47. AWB Global and Select Zone info	40
Figure 48. AE Global and Select Zone info	40
Figure 49. isp_modules.json Example	41
Figure 50. isp_sensor.json Example	42



1 ISP PQ Tuning Tools Overview

ISP PQ Tuning Tools contains two parts: tuning-server as Server and VTuner IDE as Client.

The tool allows the user to:

- Fine tune settings in ISP modules.
- Tune the sensor and ISP engine registers.
- Capture or display the output image or preview video.

1.1 Component Overview

Component	Description
Server	Executes on the board side. It receives the request from client side and controls the ISP pipeline or other modules.
Client	Executes on Windows or Ubuntu to provide a GUI environment to tune the parameters. It collects data from the server and sends control requests to the server.

1.2 Operating System Compatibility

Component	System
Server	Ubuntu 16 x64 or other Unix-like operating systems
Client	Microsoft Windows 7 x64 or later; Ubuntu 16 x64 with GUI or later

1.3 Terminology

Component	File Name
Server	tuning-server
Client	VTuner.exe

Note : Client communicates with the server via the HTTP/TCP/UART protocol.

2 Requirements

2.1 Software Requirements

Matlab Runtime MCR 2023a (for calibration tool).

2.2 Software Release Package

The software release package contains **tuning-server** and **VTuner**. Tuning-server is supported on Linux and has been verified on Ubuntu 20.04, while VTuner is supported on both Linux (with GUI) and Windows.

The release packages will have the following names:

Tuning-server:

The whole release SDK.

VTuner:

Linux: **VTuner_IDE-<version>-Linux-x86_64-<version>-Install**

Windows: **VTuner_IDE-<version>-Win32-x86_64-<version>-Setup.exe**

3 Server Usage

The Server runs on an FPGA/target board via its executable program called **tuning-server**. The Server will handle requests from the Client via Ethernet and directly control ISP behavior.

3.1 Setup Environment on FPGA

1. Startup FPGA.

Use CLI tool 't_install' to setup environment, build and debug. Before run the tool, please modify t_install/source/interact/maintain/edition/units.sh according actual needs.

Example: default preview mode are rtmp; when "TUNE_EDITION_OPTION_TRANSMIT_RTMP=0", the connection mode became tcp.

```
# export TUNE_EDITION_OPTION_DOXYGEN=1
# export TUNE_EDITION_OPTION_DRIVER_DUMMY=1
# export TUNE_EDITION_OPTION_DRIVER_UNITS=1
export TUNE_EDITION_OPTION_DRIVER_V4L2=1
export TUNE_EDITION_OPTION_XML=1
export TUNE_EDITION_OPTION_FILESYSTEM=1
export TUNE_EDITION_OPTION_OS=1
export TUNE_EDITION_OPTION_TRANSMIT_HTTP=1
export TUNE_EDITION_OPTION_TRANSMIT_RTMP=1
export TUNE_EDITION_OPTION_TRANSMIT_TCP=1
# export TUNE_EDITION_OPTION_TRANSMIT_UART=1
```

2. t_install support interactive mode as shown below:

```
./t_install/install.sh -i
```

3. Menu Tree:

```
- Build: Build application/modules, config build type
- APP: Build APP and dependencies
  - Server: Build Server and dependencies
- Module: Build single module
  - ffmpeg-5.1.2
  - opencv-4.6.0
  -openh264-2.3.1
  - t_common_c
  - t_database_c
  - t_driver_c
  - t_json_c
  - t_server_c
  - t_shell_c
  - t_transmit_c
- Build Type: Config build type
  - Debug
  - Release
- Changelog: Get changelog of all projects since when
- Debug: Use GDB to debug APP
  - Server: Debug t_server_c
- Developer Mode: Toggle developer mode for t_install
  - Off
  - On
- Edition: Change edition
  - Unify
- Environment:
```

- Prerequisite: OS
- Tools: CMake, Ninja, etc
- V4L2
 - vivid
 - Install: OS
 - Start: Load vivid driver
 - Stop: Remove vivid driver
 - Test: Launch ``xawtv`` to verify
- X Client
 - Display: Set local DISPLAY service
 - Test: Launch ``xeyes`` to verify
- Goto: Shortcut to special location
 - Root
 - Bin
- Information: Display all information of `t_install`
- Mirror: Git repo mirror operation
- Package: Pack resources
 - Full IDE
 - Minimal Source Code
 - Git Repo
- Purge: Clean files generated by build
 - Tuning
 - Dependencies
 - Tools
 - All
- Run: Launch application
 - Server
- Theme: Change theme
- Toolchain: Change toolchain for cross compiling
 - aarch64-poky-linux
 - aarch64-xilinx-linux
 - csky-linux-gnuabiv2
 - microblazeel-xilinx-linux
 - x86_64-linux-gnu
 - x86_64-w64-mingw32
- Troubleshooting: Diagnose environment variables

4. 'Environment/tools' will install CMake and Ninja tool.
5. If the RTMP video preview feature is needed, use 'Environment/nginx/install' to install the Nginx service.
6. Run 'Build/APP/Server' to build the server.

There are two video preview ways:

- a. The default RTMP method depends on H264+RTMP and Nginx on the board side, and VLC (or any other video player which supports network stream playing) on the PC side. Use 'Environment/nginx/start' to start the Nginx service. The video stream URL is "**rtmp://IP:1935/tuning/preview**".

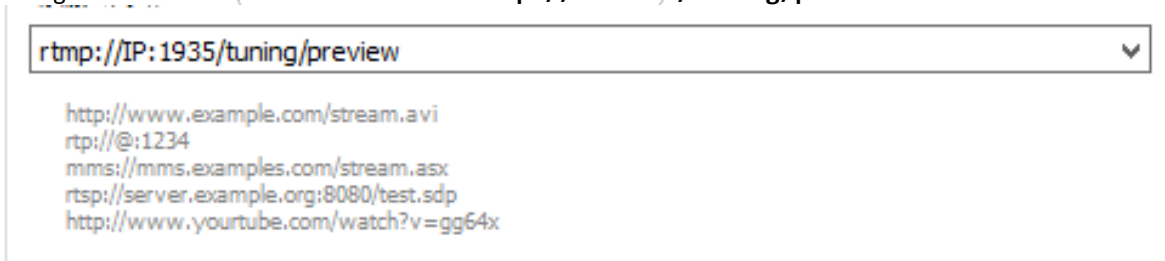


Figure 1. VLC URL

- b. The TCP method can preview the video inside the PQ Tuning Tools.

If you want to use TCP preview way, change the environment variable:

- File: `~/t_frameworks/t_install/source/interact/maintain/edition/units.sh`:
- `"TUNE_EDITION_OPTION_TRANSMIT_RTMP=1"` to `"TUNE_EDITION_OPTION_TRANSMIT_RTMP=0"`

Restart the interactive menu, compile the server again, and connect server with TCP protocol.

3.2 Run the Server

Select menu **'Run/Server'** to run the tuning server. The default HTTP port is 34567, default TCP port is 45678.



4 VTuner Usage

VTuner serves as the Client, provides the GUI to tune the parameters of ISP sub-modules.

4.1 General Steps

1. Open VTuner IDE.
2. Create an ISP project.
3. Configure Server IP and port in Server Configurations.
4. Select the feature that you want to tune in the Features view.
5. Change the parameters and check the result from preview or captured image.

4.2 Start Using VTuner IDE

4.2.1 Install and Uninstall VTuner IDE

VTuner IDE installation supports the following platforms:

- Linux
- Windows

VTuner IDE has been verified to work on Windows 10 and Ubuntu 20.04. It may work on other Windows or Linux systems but is not guaranteed.

4.2.2 VTuner IDE for Linux

4.2.2.1 Install VTuner IDE

Run `VTuner_IDE-x.x.x-Linux-x86_64-**-Install` to launch the installation wizard (x.x.x is the version number). Follow the wizard to finish installation.

4.2.2.2 Run VTuner IDE

Run `installation_dir/ide/vtuneridex.x.x` in a BASH.

4.2.2.3 Uninstall VTuner IDE

Run `installation_dir/uninstall` to launch the uninstallation wizard. Follow the wizard to finish uninstallation.

4.2.3 VTuner IDE for Windows

The installation steps on Windows and Linux are similar.

4.2.3.1 Install VTuner IDE

Run **VTuner_IDE-x.x.x-Windows-**-Setup.exe** to launch the installation wizard. Follow the wizard to finish installation.

4.2.3.2 Run VTuner IDE

- Click 'Start Menu->VeriSilicon->VTunerIDE x.x.x->VTunerIDE x.x.x' or
- double click the shortcut 'VTunerIDE x.x.x' on the desktop or
- run `installation_dir\ide\vtuneridex.x.x`

4.2.3.3 Uninstall VTuner IDE

Run `installation_dir\uninstall` to launch the uninstallation wizard. Follow the wizard to finish uninstallation.

4.2.4 Using Workspaces

During the first run of VTuner IDE, the **Workspace Launcher - Select a workspace** dialog will pop up with a default path.

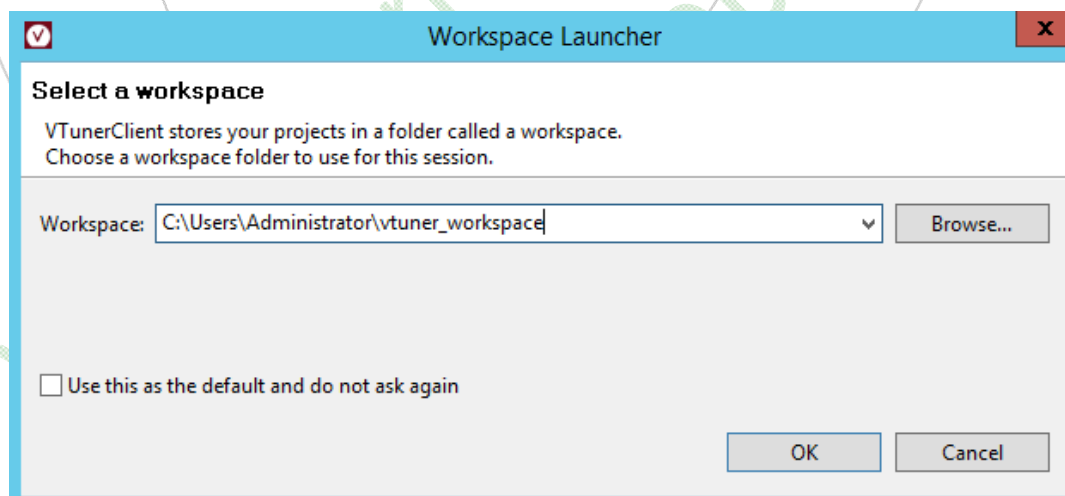


Figure 2. Select a Workspace for VTuner IDE

Click **OK**.

- The Welcome dialog will appear.

4.2.5 Switch Perspective

Switch perspective to the ISP Tuner. To do this, select **Window => Open Perspective => Other** from the main window, then choose ISP Tuner and click **OK** button, or double click **ISP Tuner**.

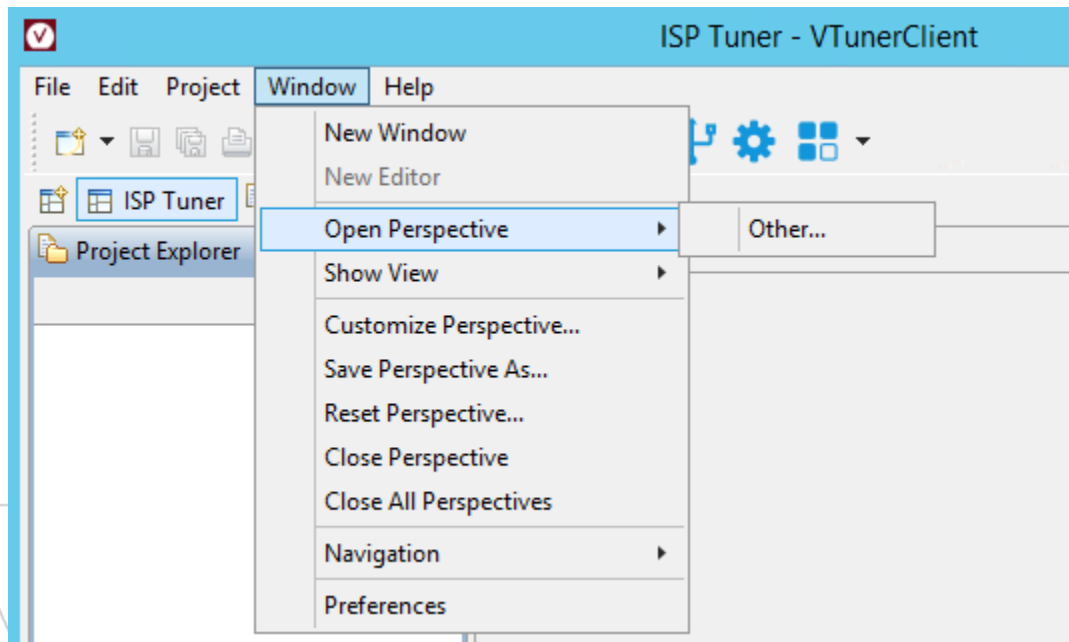


Figure 3. Open Perspective → Other

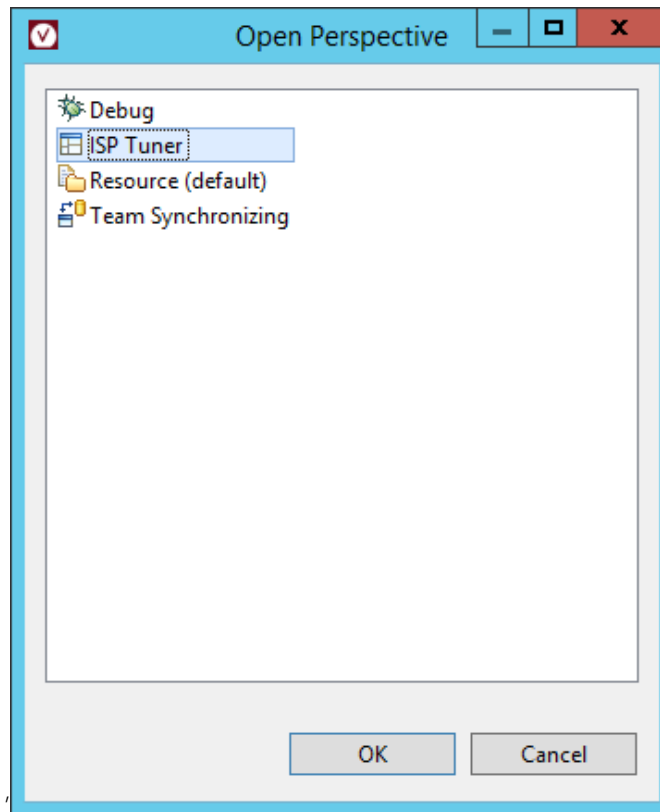


Figure 4. Open Perspective → ISP Tuner

4.2.6 Create ISP Project

To create an ISP Project, select **File=>New=>ISP Project Wizard** from the main window.

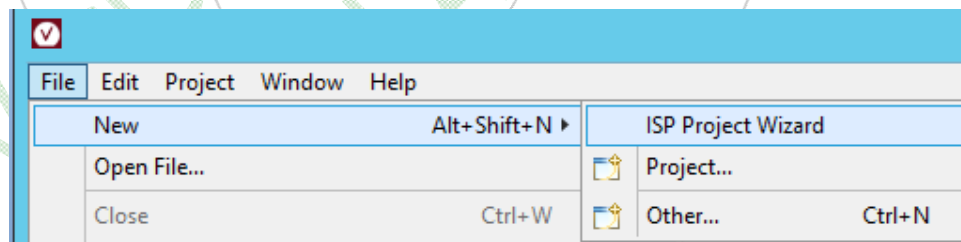


Figure 5. Select ISP Project Wizard

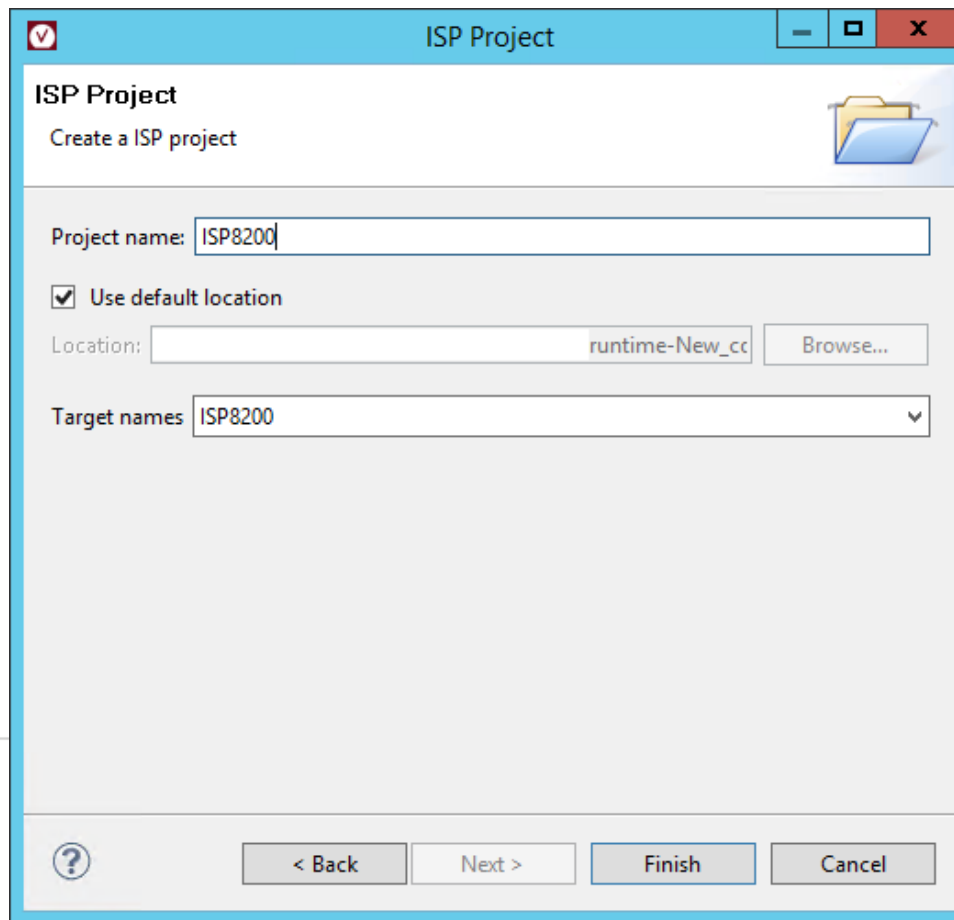


Figure 6. Create an ISP Project

Enter the Project name in the **ISP Project** dialog. For the purposes of this tutorial, "MyTest" is used as the project name.

If **Use default location** checkbox is enabled, a new directory with the same name as the project will be created in the current workspace. The path to the directory will be displayed in the "Location" field.

To select a preferred template for tuning, click the dropdown menu on the right side of Target names then click the Finish button to create the project and open the ISP Tuner perspective.

4.3 Common Features

As shown below, The ISP Tuner Perspective contains:

- A toolbar (1): Provides useful actions.
- Project Explorer (2): Explores all created projects in the current workspace.
- Features (3): Shows the template name and modules of the selected project.
- Detail View (4): Shows the parameters of the selected module.
- Console (5): Displays debug messages.

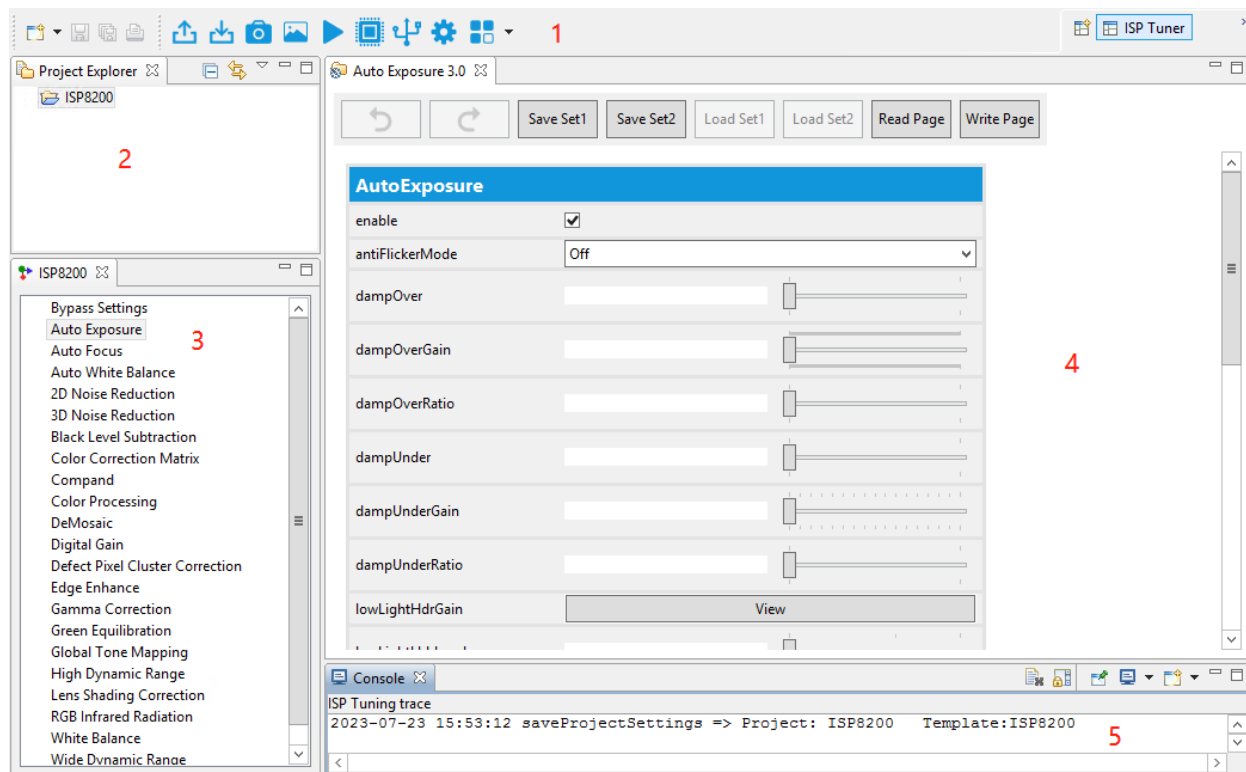


Figure 7. ISP Tuner Perspective

4.3.1 Import Tuning Data

Select a calibration JSON/XML file or a exported calibration+tuning JSON file from local path and import to server. After import done, the current page will be refreshed.

Note: Should import the calibration and default tuning data before start stream.

Select an XML or JSON file, and then click the **Open** button.

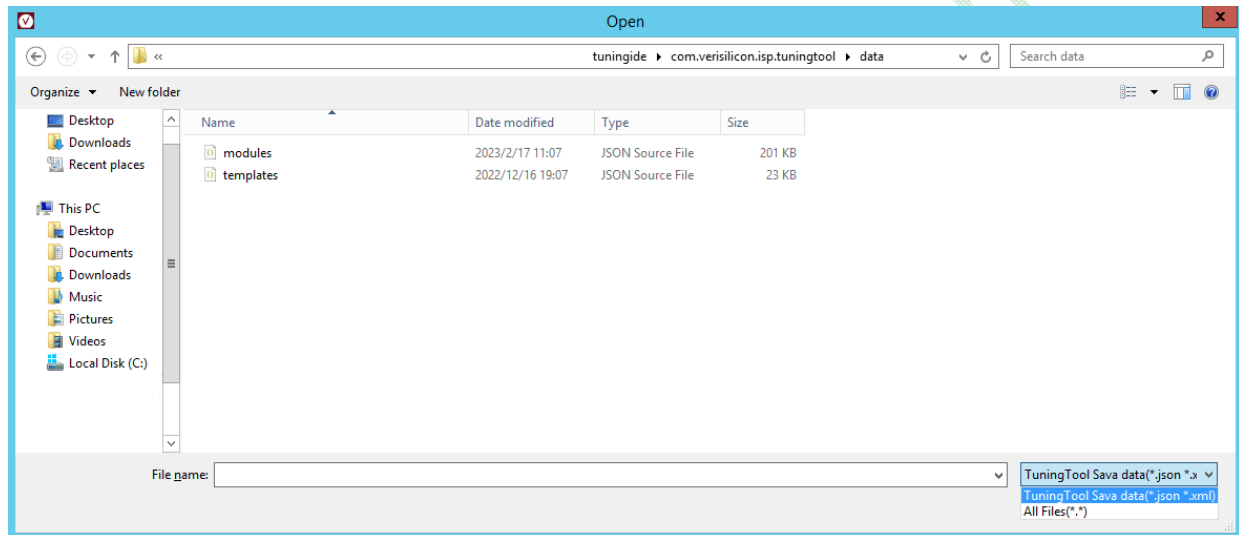


Figure 8. Select a Calibration JSON File

The Tool will show a waiting prompt while importing.

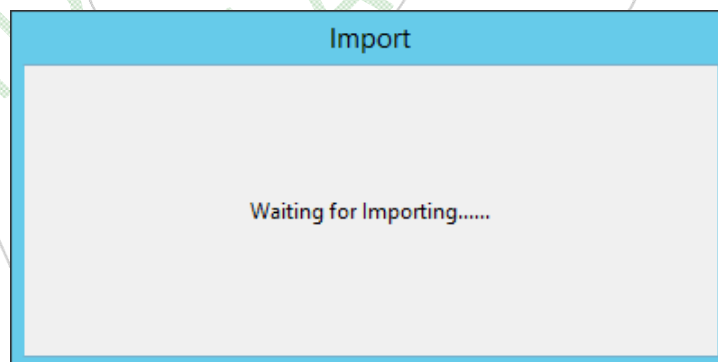


Figure 9. Popup for Waiting for Import

A message will be displayed in the dialog indicating whether the import finished successfully or not. It is the user's responsibility to ensure the correctness of the imported file.

4.3.2 Export Tuning Data

Exporting the current calibration and tuning data from the server side into a JSON file.

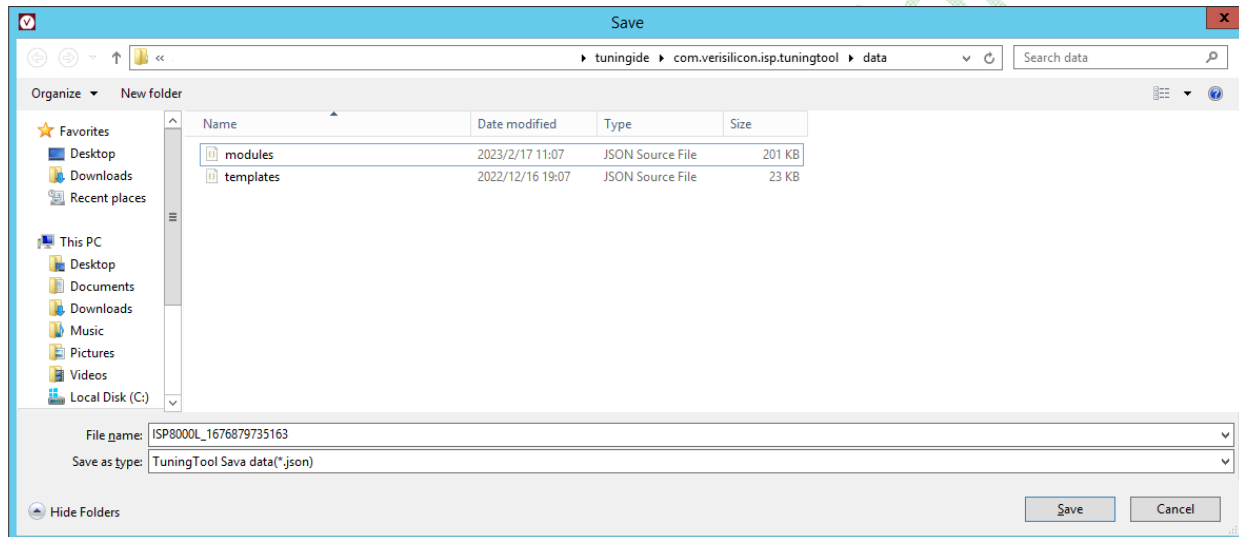


Figure 10. Export Calibration and Tuning Data

The Tool will show a waiting prompt while exporting.

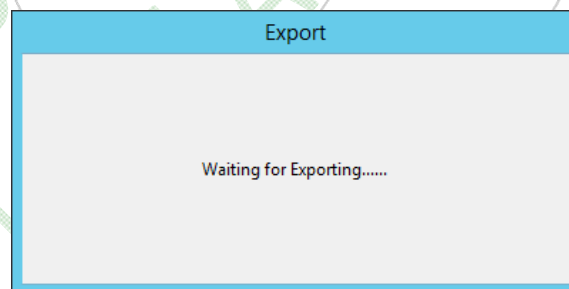


Figure 11 Popup for Waiting for Export

The user can confirm whether the exporting is successful or not via the prompt message in dialog.

4.3.3 Capture

User can specify capture image(s) or video number and save path.

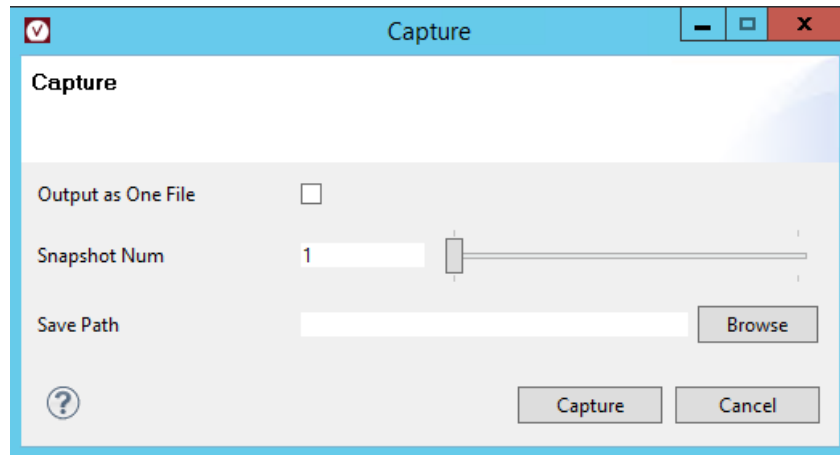


Figure 12. Configuration for Capturing Images

Image Format and size

User can choose whether to check “Output as One File” or not. If checked, a video file with the specified number of frames will be output, otherwise a sequence file with specified number of frames will be output. Image(s) or video file format and size is same as output format and size in “Live Preview” dialog.

Save Path

Specify the image saving path.

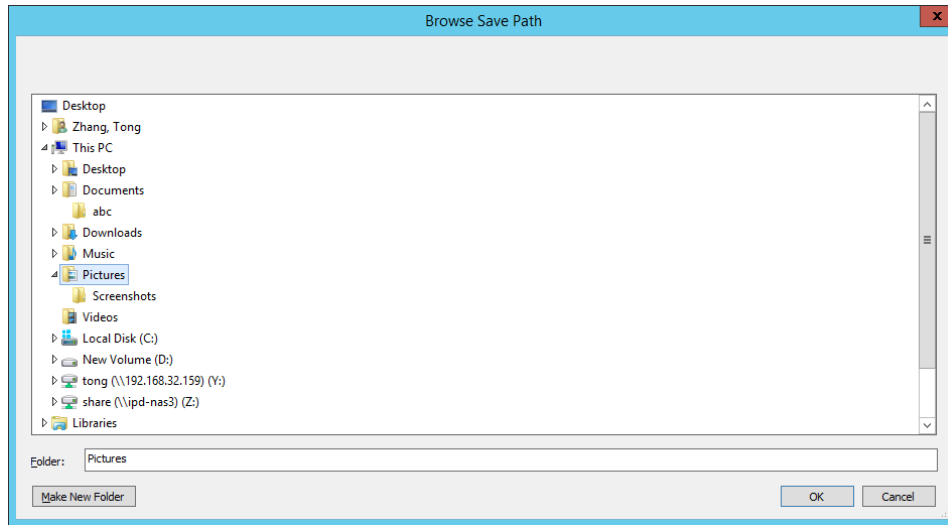


Figure 13. Configure the Save Path in Capture Settings

Snapshot Num

Capture 1~10 images. Server will return all the images together, it will take a while to get the images.

Then click **Capture** button to start capturing.

4.3.4 Raw Image Input

Send local raw image(s)/video to server for ISP processing. This must be done before start ISP pipeline.

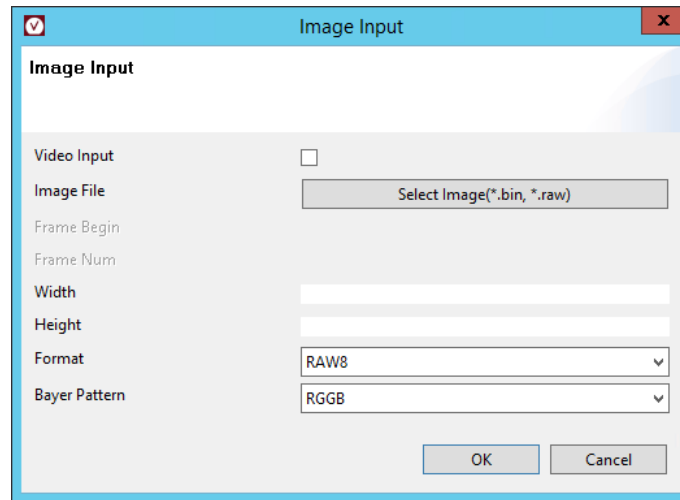


Figure 14. Image Input Dialog

Send raw image(s) to the server as the pipeline source:

- Step 1. Click the **Select Raw Image(*.bin, *.raw)** button and choose the image(s) from your local path.
- Step 2. Specify **Width**, **Height**, **Format**, **Bayer Pattern**.
- Step 3. Click the **OK** button to transfer the image data to the server for processing.
- Step 4. Capture image(s) to view process result.

Video Input

Input a raw video to the server. Enable the Video Input checkbox, then specify **Frame Begin** and **Frame Num**.

Format

Supported raw formats: RAW8, RAW10/RAW12/RAW14 with different align mode, RAW16, HDR 2/3/4DOL Image RAW12.

Bayer Pattern

RGGB, BGGR, IRBRG, RYYCY, etc.

4.3.5 Live Preview

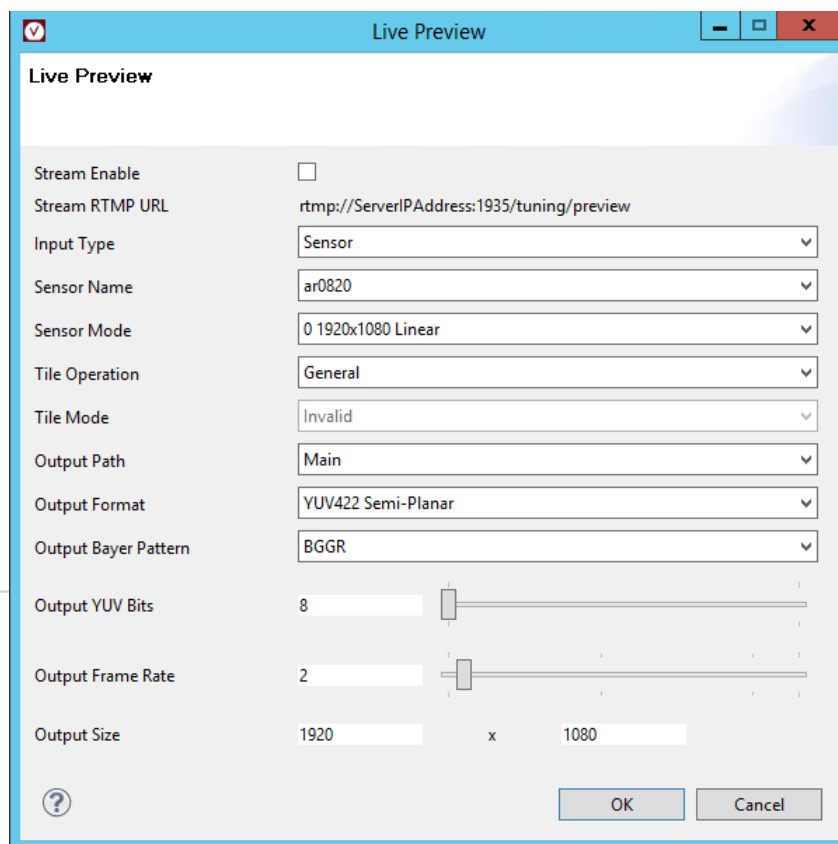


Figure 15. Live Preview Dialog

If the live stream is enabled, server will init and start the ISP pipeline, and output the preview video if output format is YUV image. According server configuration, user can preview via RTMP video player or the player inside Vtuner.

Stream Enable

Start (enable) or stop (disable) the ISP pipeline on the server.

Stream URL

Show the URL of the RTMP preview stream provided to the video player.

Input Type

Image or **Sensor**. If choose **Image**, the image(s) must have been input to server before start stream. Choose **Sensor** for live sensor tuning.

Sensor Name

Choose sensor name. Reference to “Customize Sensor” section.

Sensor Mode

Choose sensor working mode, different resolution and Linear/HDR mode.

Tile Operation

Choose tile type: **General**, **SW Split Image** and **HW Split Image**. If “**General**” is selected (which means tile is closed), **tile mode** cannot be set.

Tile Mode

Choose tile mode: **Invalid**, **2*1**, **4*3**. Users need to choose according to the actual situation of product.

Output Path

Choose pipeline output path: **Main**, **Raw**, **Self**, **Self2**. For capture raw image, should use **Raw** path as output path.

Output Format

Choose pipeline output format: **YUV**, **RGB** or **Raw**. For capture raw image, output format should be Raw. Only support preview YUV output format.

Output Bayer Pattern

For output Raw format, support **BGGR/GBRG/GRBG/RGGB**.

Output YUV Bits

For output YUV format, set bits of each color component, value range [8~14].

Output Frame Rate

ISP pipeline output frame rate, value range [1~24].

Output Size

Output size should not bigger than sensor current resolution, the width should be multiple of 16px.

4.3.6 Edit Register

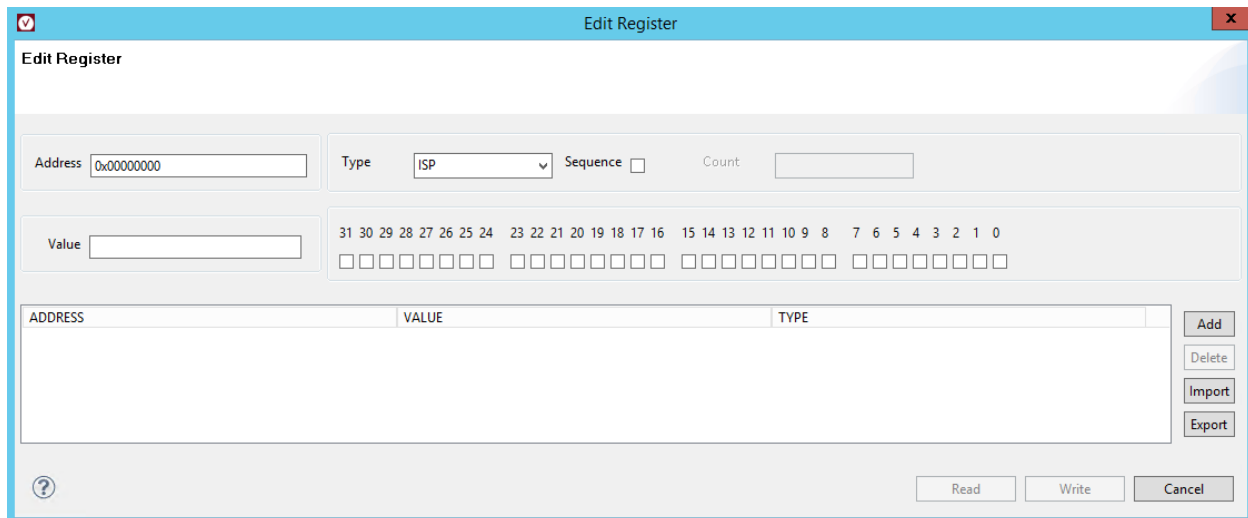


Figure 16. Edit Register

Warning: Modification of registers should completely follow the register document; otherwise, it may lead to unpredictable errors.

Address

The address of the target register.

Type

Choose **ISP** or **Sensor** address to operate.

Sequence

The user can read or write multiple registers simultaneously.

Count

Sequence registers count, which defines the number of registers to be added to the table, starting from the input of Address.

Value

Edit the value of the target register. Show and edit the value as bit mode. If a Bit checkbox is checked, the corresponding bit is set to 1; Otherwise, the bit is set to 0. The checkbox and the value will update synchronously.

Add

Add the register address and value to the table. The address cannot be empty. If the address exists in the table, adding will fail.

Delete

Delete selected items in the table.

Import

Import registers address and value from a CSV file to the table.

Export

Export items in the table to a CSV file.

Clicking on the **Read** button will read all the registers which are in the table from the server side.
Clicking on the **Write** button will write all the registers which are in the table to the server side.

Notice: There is no guarantee that all bits can be written successfully, depending on the register's actual bit width.
If the expected value is different from the actual written value, the user will receive a warning prompt.

4.3.7 Server Connection

The user can choose the server connection method using this page.

Connection Type

HTTP, TCP and UART are supported.

Host (name or IP)

Specify the host name/IP and port.

Add

Add the host and port input into the list.

Delete

Delete all selected host items from the list.

Test

Test if the selected host is accessible.

OK

Connect the selected host.

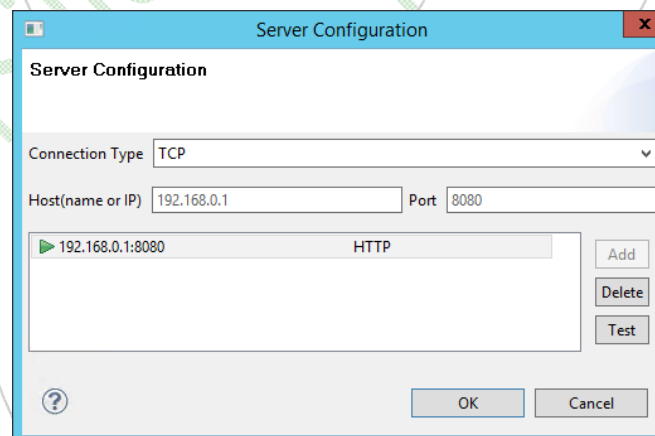


Figure 17. HTTP/TCP Server Configuration Dialog

If choose HTTP/TCP connection type, user need to enter the correct IP address and port number of the corresponding connection type.

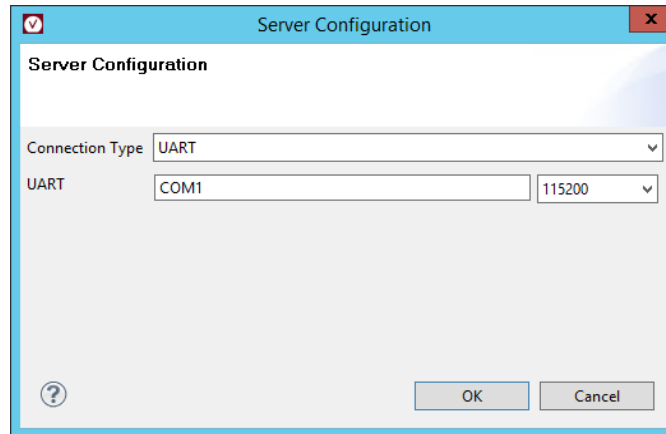


Figure 18. UART Server Configuration Dialog

If choose UART connection type, user need to enter the correct serial port and make sure user have permission to open the serial port. Baud rate should be chosen to match server side.

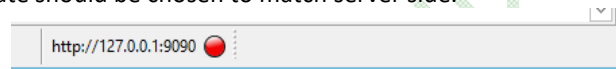


Figure 19. Light Indicating Server Connection Status

The current connection status is shown in the status bar. A green light means the connection is OK while a red light means the connection is down.

4.3.8 Preferences

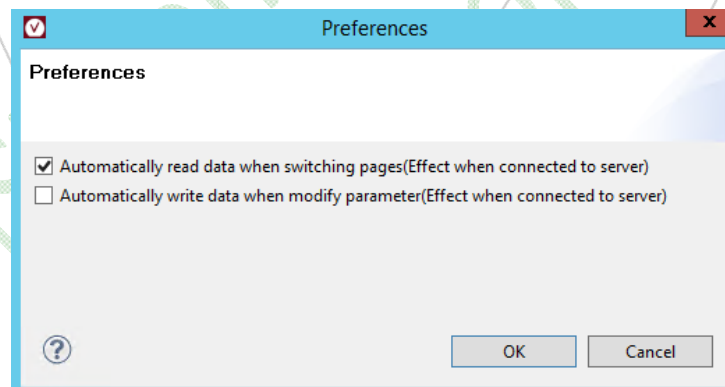


Figure 20. Preferences Setting

Auto Read Page

If checked, parameters of the current page will be read from the server when this page is shown. The default value is true.

Auto Write Page

If checked, it indicates whether it will automatically write values to the server when the values change. The default value is false.

Note: When the connection with server for the first time is ready, user should read the page manually.

4.3.9 Extra Component

The ISP PQ Tuning Tools provides extra components.

JSON/XML edit

A simple JSON/XML file editor.

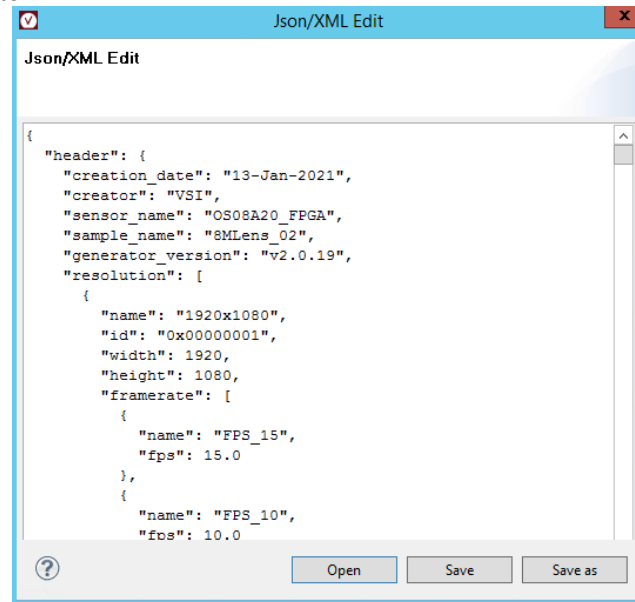


Figure 21. JSON/XML Edit

Calibration Tool

Start the Calibration Tool. This tool requires MCR2023a installation. Please refer to the ISP Calibration Tools document.

AWB/AE Tool

Show the AWB/AE analyze tool, more detail refer to the AWB/AE Tool chapter.

4.3.10 Project Explorer

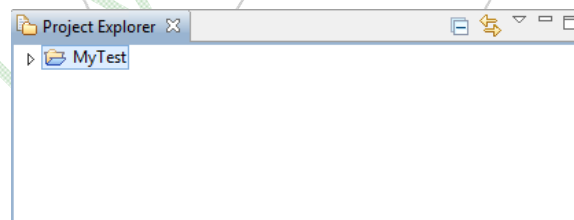


Figure 22. Project Explorer

The user can create projects for each template.

Select an ISP project. All supported features of the current target will be shown in the Features view.

4.3.11 Features View

Features View shows the supported features of the current project.

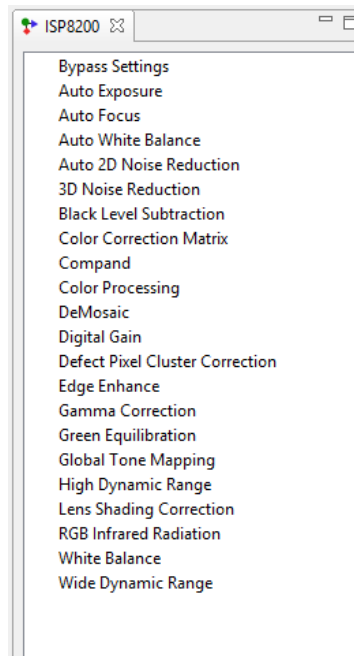


Figure 23. Features View for the Current Project

Click the feature item, the feature details will be shown in the **Detail View**.

4.3.12 Detail View

This page shows a control bar and the parameters for the selected feature.

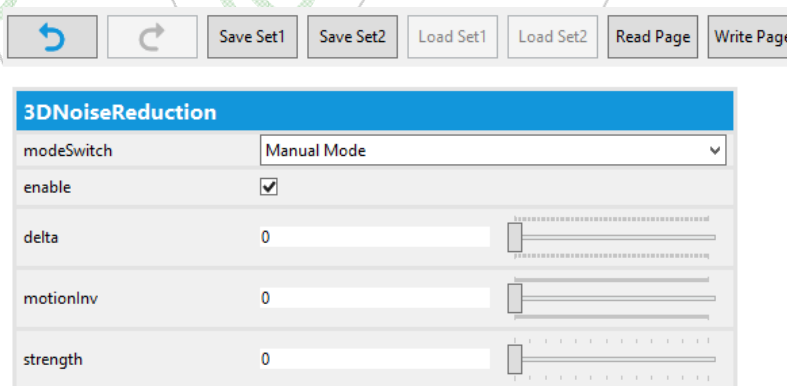


Figure 24. Detail View

a. Control Bar

There are several actions in the Control Bar: Undo, Redo, Save Set, Load Set, Read Page, Write Page.

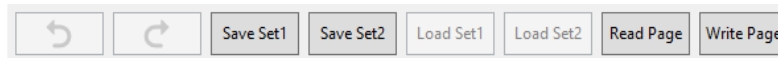


Figure 25. Control Bar



Please note that the undo and redo function is supported for each module individually, allowing up to 10 steps to be saved.

Any new behavior will be stored at the end of the history, with intermediate steps if necessary. For example, the initial value of **Strength** is 0, and is set sequentially to 2, 4, 6, and 8. Clicking the **Undo** button repeatedly until the value is set back to 2 will enable both the undo and redo button, as we can track backward to 0 and forward to 4, 6, and 8.

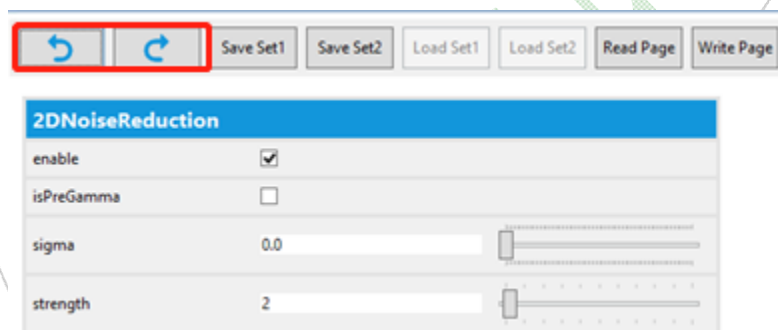


Figure 26. Back to 2, Undo Button and Redo Button is Enabled

If a new value is set for **Strength** at this point, e.g., 5, it would be added at the end of the record, after an intermediate step 8=>2. In this scenario, the redo button will be disabled since the last step has been reached, and the **Undo** button can set the Strength back as: 5=>2=> 8=> 6=> 4=> 2=> 0.

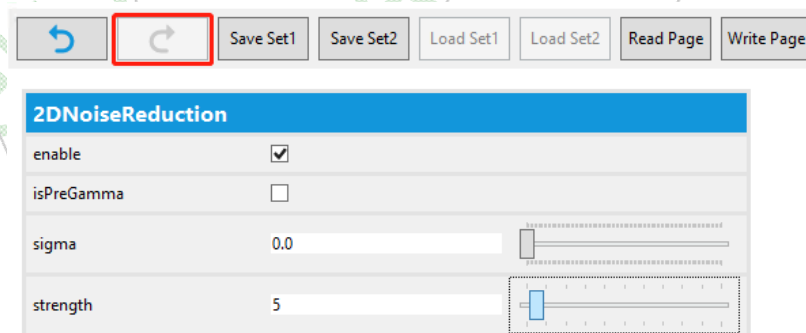


Figure 27. Input 5, Redo Button is Disabled

Save Set1

Save a set of current page parameters temporarily.

Save Set2

Save another set of current page parameters temporarily.

Load Set1

Load parameters from set1.

Load Set2

Load parameters from set2.

Read Page

Read all the parameters of the current page from server.

Write Page

Write all the parameter values of the current page to server.

- Parameter Details: All parameters for the selected feature are shown here. The interfaces and controls vary depending on the type of parameters. Please refer to the Data section for details.

4.3.13 Console

The **Console** section displays all interaction messages between Client and Server, and other prompt messages.

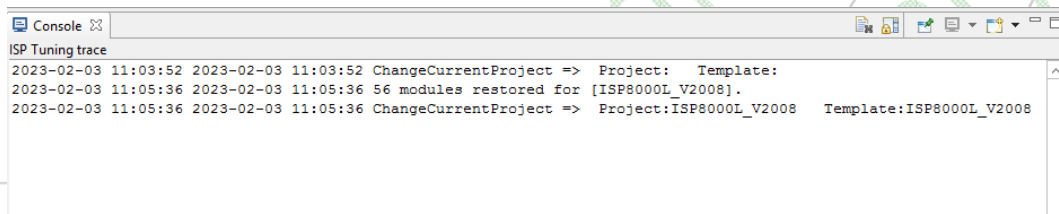


Figure 28. Console with Messages for Selecting a Project

4.4 Parameter Data Types

Currently, there are five types of parameters supported:

1. Boolean
2. Integer or float value
3. Option
4. Matrix
5. Curve

The data type specifies the controlling and the interface for the parameter in Detail View.

4.4.1 Boolean Value

Edit such value with a checkbox.

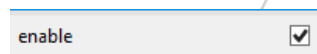


Figure 29. Boolean Value in Detail View

4.4.2 Integer or Float Value

Edit such value with a text input and a slider, which will update synchronously. Value can't be set out of its range.



Figure 30. Integer or Float Value

4.4.3 Option Value

Option parameters are shown in a combo box.



Figure 31. Option Value

If modules have the “Mode Switch” option, the user can switch between manual and auto control page.

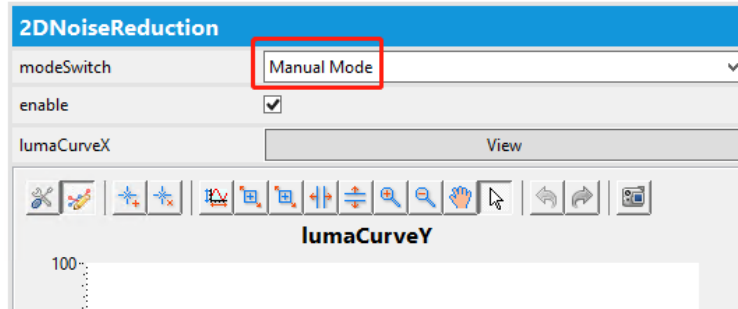


Figure 32. Manual UI

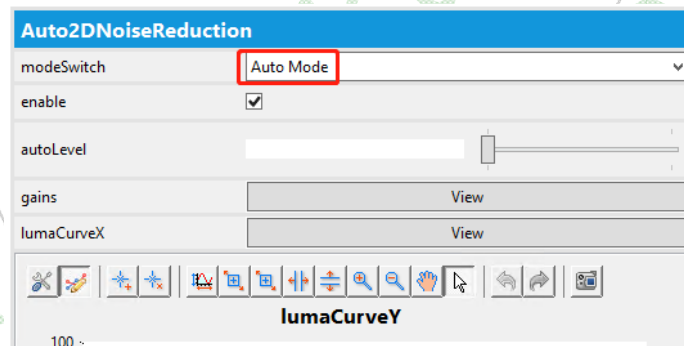


Figure 33. Auto UI

4.4.4 Matrix Value



Figure 34. Matrix Value

There are two types of matrix value, including number and roi.

4.4.4.1 Number Matrix Value

Click View to edit with a table. The first row of the table consists of names of columns. Any edit of the table can be saved only after pressing the **OK** button. Support importing a .csv file into the table or exporting the current table into a .csv file. User should ensure the correction of the file for importing.

The table is paging display, user can switch to previous/next page by click the arrow buttons. Valid value range will be shown in the title.

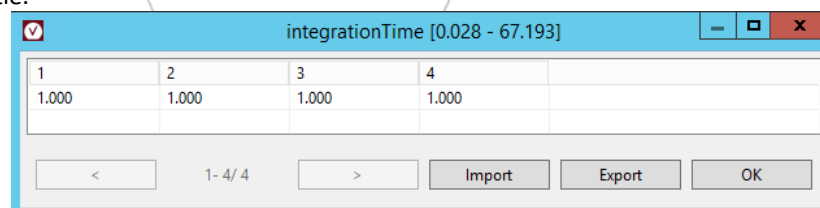


Figure 35. Table for Editing Number Matrix Value

4.4.4.2 ROI Matrix Value

Click View to edit with a table and an image. User can select regions of interest on the image. The table will add information such as coordinates, width, height and weight of corresponding ROI. User also can click “del” button in the table to delete selected ROIs. Any edit of the table can be saved only after pressing the OK button.

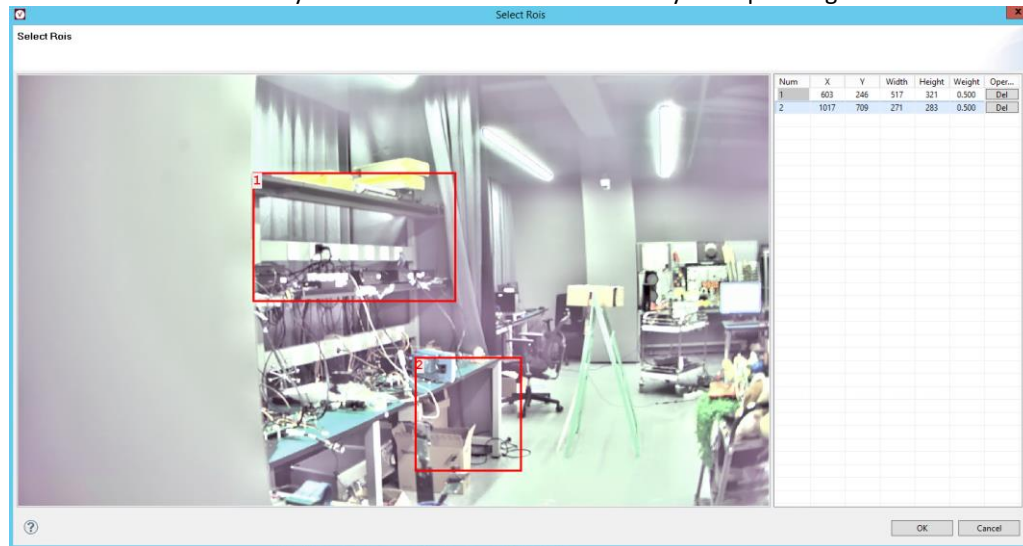


Figure 36. Table for Editing ROI Matrix Value

4.4.5 Curve Value

Parameters with the curve type are shown as a line chart and a two-dimensional matrix table. Each control point on the chart represents a value at a specific row and column of the matrix. The matrix can be edited by either dragging control point on the chart or modifies the corresponding value in the matrix table. The number of control points can be modified, min value is 3, max value equal to the number of X coordinates. Control point number only affect curve display, not affect the actual X coordinates count of curve.

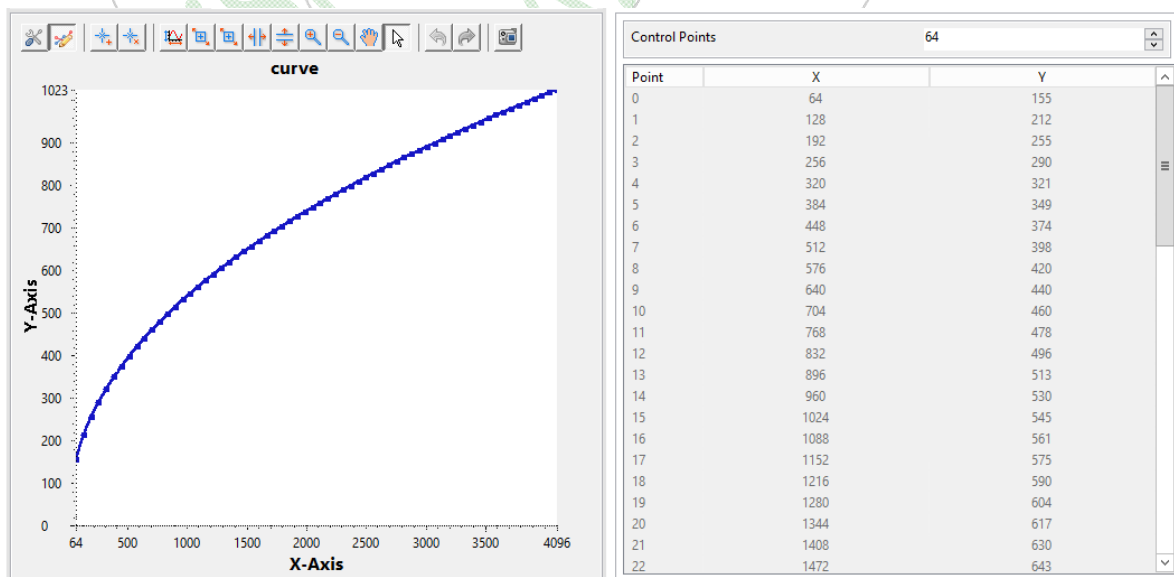
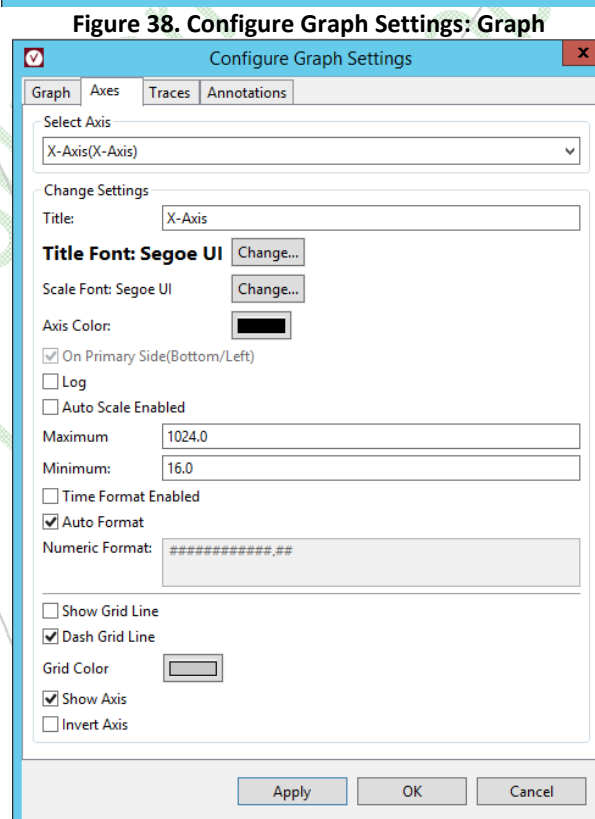
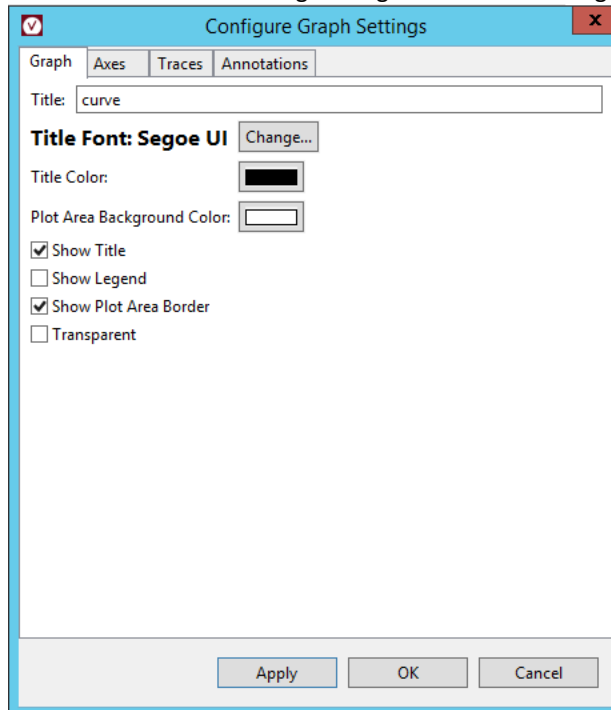


Figure 37. Curve Value: a Curve and a Table

There is a toolbar on top of the chart. The main features of the toolbar are:

Configure Settings

Configure the appearance of the chart with the following settings as shown in the figure below.



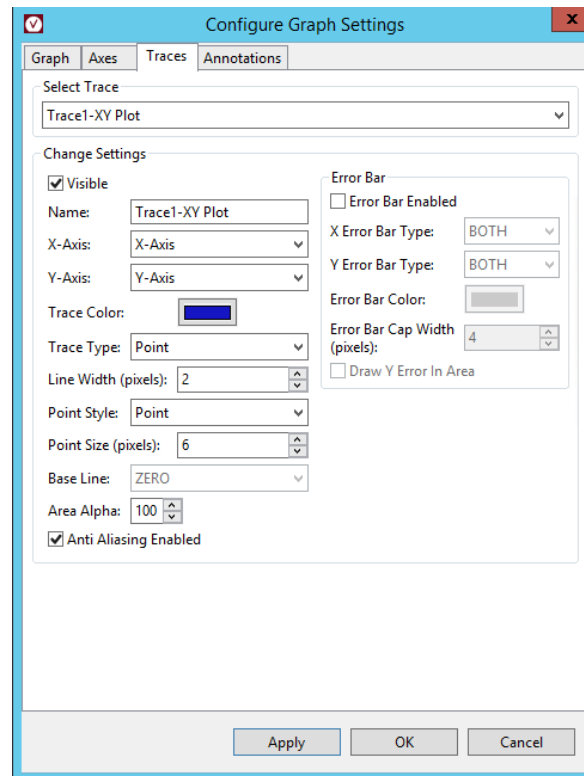


Figure 40. Configure Graph Settings: Traces

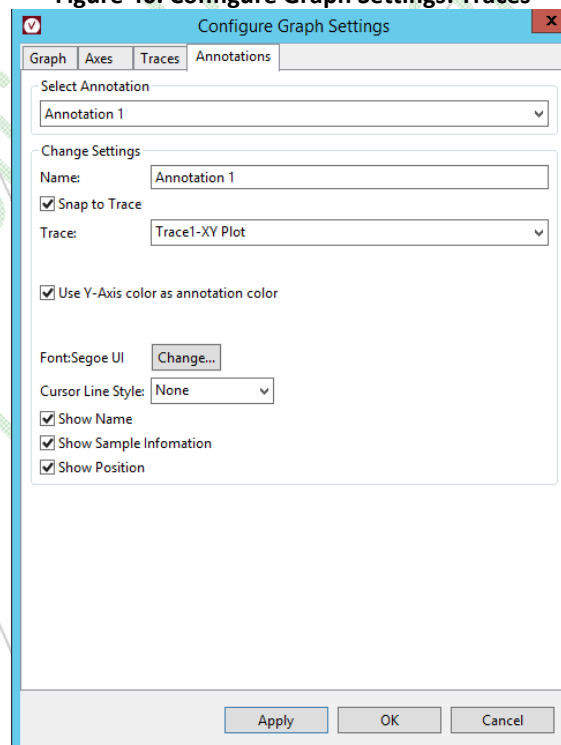


Figure 41. Configure Graph Settings: Annotation

Show Legend

Toggle to show the legend when the legend is available.

Add/Remove Annotation

Add or remove annotations on the chart. An example of two annotations on the chart are shown in the figure below.

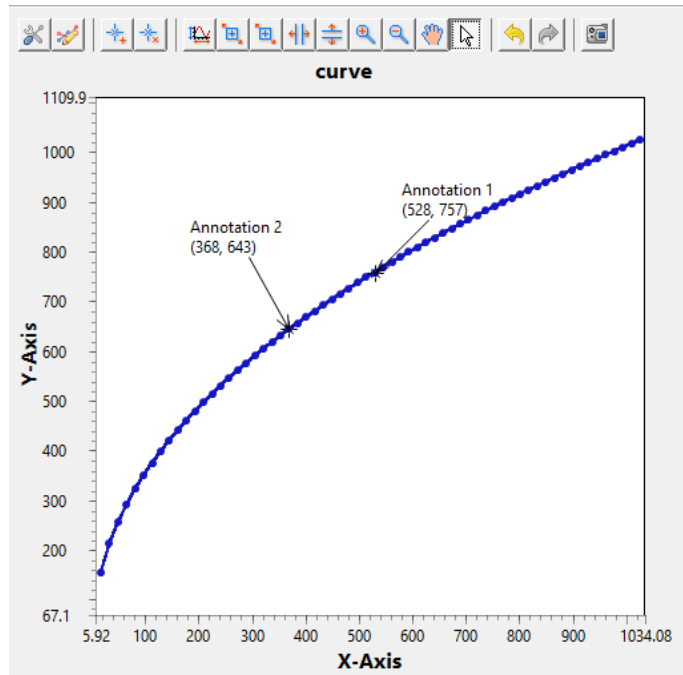


Figure 42. Graph: Annotation

Scale/Zoom

Perform auto scale and zoom operations.

Panning

Move (pan) the chart.

Undo/Redo

Undo or redo the previous operations.

Save Snapshot

Save a snapshot to a PNG file.

4.4.6 Color Matrix

Color matrix shows statistics value in a grayscale table, the label in the cell is the statistics value, 0 is shown in black and 255 is shown in white. For exposure statistics, the matrix size is 25x25, 32x32, or other size, depending on the hardware statistics.

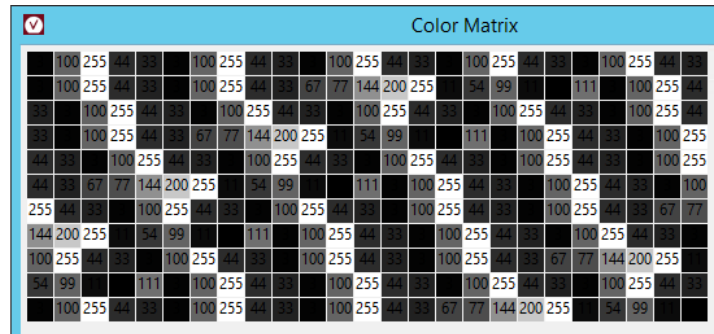


Figure 43. Graph: Annotation

4.5 AWB/AE Tool

4.5.1 AWB Tool

AWB tool is used to visualize the corresponding statistical information, by selecting **AWB (AE) Tool** in the Extra Component in the menu.

Before use these tools, client should have established connection with server and started stream.

AWB tool can plot the calibration data, as well as the statistical data such as the average RGB value of (valid) white points, the proportion of white points that located in each color temperature range, etc.

The AWB calibration data used to generate the curve and capture an image automatically. Meantime, statistic data (an array with 32*32 elements that of double value type) will be read from server.

The captured image will be displayed in a new window, covered with block_size * block_size (32 * 32) rectangles. Each rectangle represents an element in the statistics data. User can select a specified rectangle to get the local information of the selected zone. The selected zone will be displayed with a red border. Only one zone can be selected at same time, and user can click a selected (red) zone to cancel the selection.

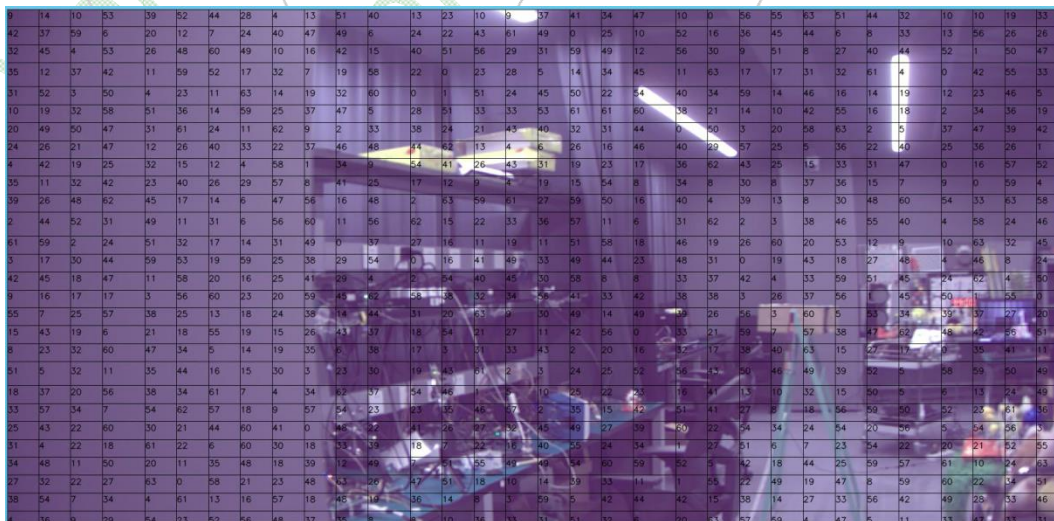


Figure 44. Captured Image and 32 * 32 Blocks

9	14	10	53	39	52
42	37	59	6	20	12
32	45	4	53	26	48
35	12	37	42	11	59

Figure 45. Select a Block

There will be another window contains a table and a chart. Both the global and the local (selected zone) statistic information will be displayed in the table, including the (average) value of each channel of the image data. If there is no selected zone, values for the selected zone will be empty. "Global" in table represents the mean of white point in R, G and B channels that participate in AWB calculation effectively. "SelectZone" in table represents the mean of three channels of R, G and B in selected zone.

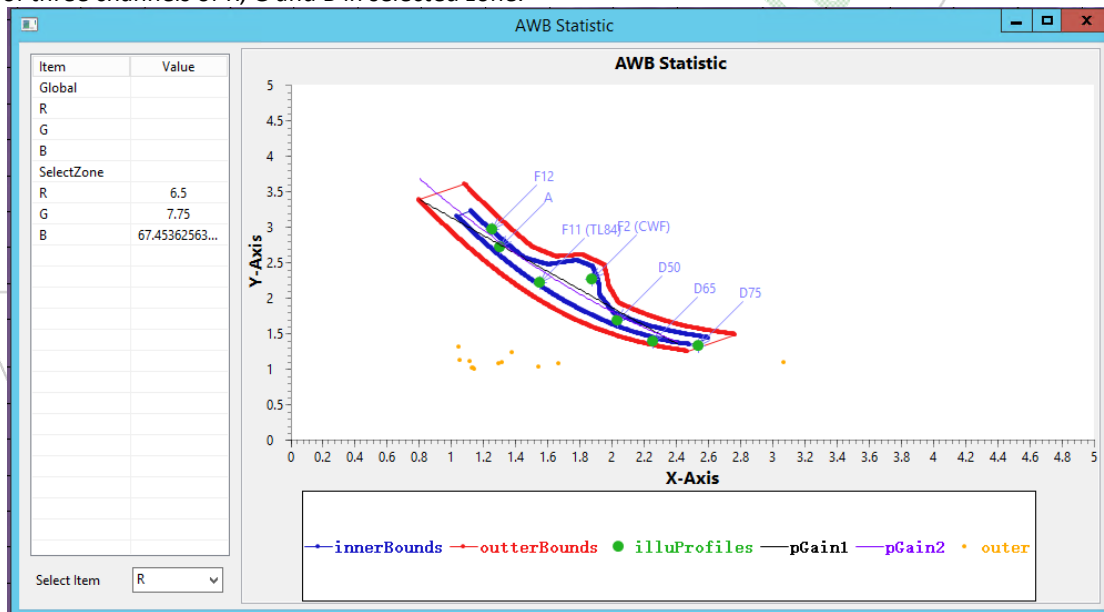


Figure 46. AWB Statistics Info and Chart

When a block is selected, and a channel at the bottom of the table is selected, the **SelectZone** in the table shows the information for the selected block (the red block in Figure 45). In blocks window, each block will show the selected channel value.

Client side will read data from server every 25 seconds and refresh captured image (Figure 44), chart and table (Figure 46).

Figure 47. AWB Global and Select Zone info

4.5.2 AE Tool

[illegible]

Figure 48. AE Global and Select Zone info

4.6 Data Sync with Server

Connect to the server before changing the parameter value. When a connection is established, the client will not sync data automatically, so the user should sync data manually, unless the user switches to another module. In this case, the client will automatically read data from the server if the auto read feature is enabled in **Preferences Dialog** (Figure 20). Any modification of the value will be sent to the server automatically if **auto write is enabled in the Preferences Dialog** (Figure 20).

After importing the JSON file to the server, the client will read the current page from the server automatically.

4.7 Customize Sub-module and Parameters

There is a JSON file named **isp_modules.json** under workspace path defines all sub-modules and parameters displayed on the UI. This JSON file may be modified when you have the following requirements:

- Remove certain modules or parameters
- Modify the display order of modules or parameters
- Modify the module, group or parameter display name
- Combine some parameters into a new module
- Collaborate with the server to add modules or parameters

```
{
  "moduleName": "Auto DeNoising PreFilter",
  "paramGroups": [
    {
      "groupName": "AutoDeNoisingPreFilter",
      "params": [
        {
          "category": "ADPF",
          "key": "enable",
          "type": "toggle",
          "value": {
            "current": null
          }
        },
        {
          "category": "ADPF",
          "key": "tableIndex",
          "label": "autoLevel",
          "type": "range",
          "value": {
            "current": null,
            "decimals": 1,
            "max": 1000,
            "min": 0
          }
        }
      ]
    },
    {
      "category": "ADPF",
      "key": ":div",
      "label": "div",
      "type": "matrix",
      "value": {
        "current": null,
        "decimals": 0,
        "max": 64,
        "min": 0,
        "relateTable": "gains"
      }
    }
  ]
}
```

Figure 49. isp_modules.json Example

"moduleName" defines the name of the module.

"paramGroups" define parameter groups within a module.

The “**params**” array is a set of parameters within a group, where the “**label**” of the param is the parameter name on the UI (if no “**label**”, the UI will show “**key**” instead), and the “**type**” defines the parameter’s type. Supported types including range, option, matrix, and so on.

Notes:

- After modifying the JSON file, the projects must be deleted and recreated to take effect.

Backup the JSON file before modify it.

4.8 Modify Sensor Work Mode

There is a JSON file named **isp_sensor.json** under workspace which defines all sensors work mode. Attributes of a work mode including **bayerPattern**, **bitWidth**, **fps**, **hdr**, **size**, **integrationTime**(in ms), **aGain** range and **dGain** range..

Here is a **isp_sensor.json** example:

```
[
  {
    "sensorName": "ar0820",
    "modes": [
      {
        "af": null,
        "bayerPattern": "GRBG",
        "bitWidth": 12,
        "fps": 15,
        "hdr": "Linear",
        "size": {
          "height": 1080,
          "width": 1920
        },
        "integrationTime": {
          "max": 67.193,
          "min": 0.029
        },
        "aGain": {
          "max": 8,
          "min": 1
        },
        "dGain": {
          "max": 1,
          "min": 1
        }
      },
      {
        "af": null,
        "bayerPattern": "GRBG",
        "bitWidth": 12,
        "fps": 6.7,
        "hdr": "Linear",
        "size": {
          "height": 2160,
          "width": 3840
        },
        "integrationTime": {
          "max": 149.312,
          "min": 0.064
        },
        "aGain": {
          "max": 8,
          "min": 1
        },
        "dGain": {
          "max": 1,
          "min": 1
        }
      }
    ]
  }
]
```

Figure 50. **isp_sensor.json** Example

4.9 Customize TCP Timeout Setting

TCP timeout setting defined in **isp_config.ini** under workspace. User can change the timeout according actual network situation.

tcp.controlTimeout

Defines the timeout of command when using TCP connection. Time unit is second.

tcp.streamTimeout

Defines the timeout of preview stream when using TCP connection. Time unit is second.

Document Revision History

This section describes differences between document revisions.

Note: This document is not necessarily updated for each patch or minor revision.

Document Revision	Date	Compatible product	Notes
1.00	2023-07-31	ISP8000 Series IP	Initial
1.10	2023-10-11	ISP8000/8200 Series IP	Update